

## Graduado en Matemáticas e Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Programación y simulación de la comunicación  
intercelular mediante conjugación en el simulador  
multicelular GRO

Autor: Irene Serrano Sebastián

Director: Alfonso Rodríguez-Patón Aradas

MADRID, JUNIO DE 2014



## ÍNDICE

<b>RESUMEN .....</b>	<b>V</b>
<b>ABSTRACT .....</b>	<b>VI</b>
<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 INTRODUCCIÓN.....	1
1.2 OBJETIVOS.....	2
1.3 PLAN DEL TRABAJO.....	3
<b>CAPÍTULO 2. ESTADO DEL ARTE.....</b>	<b>5</b>
2.1 OTROS SIMULADORES.....	5
<b>CAPÍTULO 3. FUNDAMENTOS.....</b>	<b>9</b>
3.1 PLÁSMIDOS.....	9
3.2 CONJUGACIÓN .....	10
<b>CAPÍTULO 4. SIMULADOR GRO.....</b>	<b>13</b>
4.1 SIMULADOR GRO.....	13
4.2 MOTOR FÍSICO CELLENGINE .....	15
<b>CAPÍTULO 5. NUEVAS FUNCIONALIDADES .....</b>	<b>17</b>
5.1 DONANTES, RECEPTORAS Y TRANSCONJUGADAS.....	17
5.1.1 <i>Datos experimentales</i> .....	18
5.2 CONJUGACIÓN .....	20
<b>CAPÍTULO 6. RESULTADOS Y CONCLUSIONES .....</b>	<b>23</b>
6.1 SIMULACIONES.....	23
6.1.1 <i>Versión Simplificada</i> .....	24
6.1.2 <i>Versión Completa</i> .....	26
6.2 RESULTADOS.....	28
6.2.1 <i>Versión Simplificada</i> .....	28
6.2.2 <i>Versión Completa</i> .....	31
6.3 CONCLUSIONES.....	34
6.4 TRABAJOS FUTUROS .....	35
<b>REFERENCIAS .....</b>	<b>36</b>

## ÍNDICE DE FIGURAS

ILUSTRACIÓN 1. PLÁSMIDO CONJUGATIVO EXTRACROMOSÓMICO EN EL INTERIOR DE LA CÉLULA PROCARIOTA.....	10
ILUSTRACIÓN 2. PROCESO DE CONJUGACIÓN OBTENIDO POR MICROSCOPIA ELECTRÓNICA.....	11
ILUSTRACIÓN 3. PROCESO DE CONJUGACIÓN.....	11
ILUSTRACIÓN 4: CRECIMIENTO DE UNA COLONIA BACTERIANA EN GRO.....	14
ILUSTRACIÓN 5. DEFINICIÓN DE UNA CÉLULA EN CELLENGINE.....	15
ILUSTRACIÓN 6. CÁLCULO DE SOLAPAMIENTOS.....	16
TABLA 1. TIEMPO DE GENERACIÓN.....	18
TABLA 2. TASA DE CRECIMIENTO.....	18
ILUSTRACIÓN 7. NÚMERO DE BACTERIAS RESPECTO DEL TIEMPO.....	19
ILUSTRACIÓN 8. RELACIÓN $T/(T+D)$ .....	19
ILUSTRACIÓN 9. HERENCIA ECOLI.....	21
ILUSTRACIÓN 10. AURA PARA EL CÁLCULO DE BACTERIAS VECINAS.....	22
ILUSTRACIÓN 11. SIMULACIÓN VERSIÓN SIMPLIFICADA.....	25
ILUSTRACIÓN 12. SIMULACIÓN VERSIÓN COMPLETA.....	27
TABLA 3. DATOS VERSIÓN SIMPLIFICADA.....	29
ILUSTRACIÓN 13. NÚMERO DE BACTERIAS VERSIÓN SIMPLIFICADA.....	29
ILUSTRACIÓN 14. $T/(T+D)$ VERSIÓN SIMPLIFICADA.....	30
TABLA 4. DATOS VERSIÓN COMPLETA.....	32
ILUSTRACIÓN 15 NÚMERO DE BACTERIAS VERSIÓN COMPLETA.....	32
ILUSTRACIÓN 16. $T/(T+D)$ VERSIÓN COMPLETA.....	33

# *Resumen*

Los conjuntos bacterianos son sistemas dinámicos difíciles de modelar debido a que las bacterias colaboran e intercambian información entre sí. Estos microorganismos procariotas pueden tomar decisiones por mayoría e intercambiar información genética importante que, por ejemplo, las haga resistentes a un antibiótico. El proceso de conjugación consiste en el intercambio de un plásmido de una bacteria con otra, permitiendo así que se transfieran propiedades.

Estudios recientes han demostrado que estos plásmidos pueden ser reprogramados artificialmente para que la bacteria que lo contenga realice una función específica [1]. Entre la multitud de aplicaciones que supone esta idea, el proyecto europeo PLASWIRES está intentando demostrar que es posible usar organismos vivos como computadores distribuidos en paralelo y plásmidos como conexión entre ellos mediante conjugación. Por tanto, mediante una correcta programación de un plásmido, se puede conseguir, por ejemplo, hacer que una colonia de bacterias haga la función de un antibiótico o detecte otros plásmidos peligrosos en bacterias virulentas.

El proceso experimental para demostrar esta idea puede llegar a ser algo lento y tedioso, por lo que es necesario el uso de simuladores que predigan su comportamiento. Debido a que el proyecto PLASWIRES se basa en la conjugación bacteriana, surge la necesidad de un simulador que reproduzca esta operación.

El presente trabajo surge debido a la deficiencia del simulador GRO para reproducir la conjugación. En este documento se detallan las modificaciones necesarias para que GRO pueda representar este proceso, así como analizar los datos obtenidos e intentar ajustar el modelo a los datos obtenidos por el Instituto de Biomedicina y Biotecnología de Cantabria (IBBTEC).

# *Abstract*

Bacterial colonies are dynamical systems difficult to model because bacteria collaborate and exchange information with each other. These prokaryotic organisms can make decisions by majority and exchange important genetic information, for example, make them resistant to an antibiotic. The conjugation process is the exchange of a plasmid from one bacterium to another, allowing both to have the same properties.

Recent studies have shown that these plasmids can be artificially reprogrammed to make the bacteria that contain it to perform a specific function [1]. Among the multitude of applications involved in this idea, the European project PLASWIRES is attempting to prove that it is possible to use living organisms as parallel and distributed computers with plasmids acting as connectors between them through conjugation. Thus, by properly programming a plasmid, you can get a colony of bacteria that work as an antibiotic or detect hazardous plasmids in virulent bacteria.

The experimental process to prove this idea can be slow and tedious, so the use of simulators to predict their behavior is required. Since PLASWIRES project is based on bacterial conjugation, a simulator that can reproduce this operation is required.

This work arises due to the absence of the conjugation process in the simulator GRO. This document details the changes made to GRO to represent this process, analyze the data and try to adjust the model to the data obtained by the Institute of Biomedicine and Biotechnology of Cantabria ( IBBTEC ).

This project has two main objectives, the first is to add the functionality of intercellular communication by conjugation to the simulator GRO, and the second is to use the experimental data obtained by the IBBTEC. To do this, the following points should be followed:

- Study of conjugation biology as a mechanism of intercellular communication.
- Design and implementation of the algorithm that simulates conjugation.
- Experimental validation and model adjust to the experimental data on rates of conjugation and bacterial growth.

# *Capítulo 1*

## *INTRODUCCIÓN*

---

1.1	INTRODUCCIÓN.....	1
1.2	OBJETIVOS.....	2
1.3	PLAN DEL TRABAJO.....	3

---

### **1.1 Introducción**

La biología sintética consiste, tanto en diseñar y construir nuevos dispositivos y organismos biológicos artificiales, como en modificar y reprogramar los sistemas biológicos naturales. El objetivo primordial de la biología sintética es hacer diseños biológicos más eficientes para poder usar estos organismos modificados en la resolución de problemas.

Es un campo emergente y desconocido donde no es sencillo prever el comportamiento de estos nuevos organismos artificiales, y comprobar dicho comportamiento mediante experimentos puede suponer un gran coste de tiempo por lo lentos que pueden ser ciertos procesos. Por esta razón, es necesario acudir a la Biología de Sistemas.

La Biología de Sistemas se encarga de reproducir el comportamiento de sistemas biológicos. Se basa en la construcción de modelos matemáticos en los que se representan las interacciones con elementos externos e internos y ayudan al entendimiento de sus procesos biológicos. Con estos modelos se pueden programar



simuladores y así predecir el comportamiento de ciertos sistemas sin tener que esperar a que ocurran experimentalmente.

A pesar de la diversidad de trabajos realizados en este tema, las comunidades microbianas siguen siendo difíciles de modelar y simplificar debido a la multitud de procesos que ocurren entre ellas. Este proyecto trata sobre estas comunidades, especialmente en modelar uno de los procesos que ocurren entre los sistemas bacterianos, la comunicación bacteriana por medio de la conjugación. Para el modelaje de este proceso será necesario basarse en uno de los simuladores que actualmente reproducen en cierta medida el comportamiento de comunidades bacterianas.

Un método común para capturar como las interacciones de bajo nivel entre las células dan lugar a comportamientos a nivel de población es con modelos basados en el individuo, AbM/IbM (Agent based Model/ Individual based Model). Estos modelos consideran poblaciones de agentes autónomos, cada uno siguiendo una serie de reglas internas que interactúan entre sí en un entorno común.

Actualmente, la gran mayoría de los simuladores que reproducen el comportamiento de colonias de bacterias se centran en el crecimiento y las reacciones a sustancias químicas. Entre estos simuladores existen varios tipos en relación a la forma de la bacteria, los que las ilustran en forma de bacilo (capsular) o en forma de coco (esférica). Dado que este proyecto se centra en la bacteria E. Coli, la cual tiene forma capsular, será necesario un simulador que contemple esta forma.

GRO es un simulador basado en el modelo AbM/IbM y que permite definir numerosos parámetros sobre el comportamiento de una colonia, además de poseer la capacidad de representar la expresión de proteínas, factor importante en el campo de los circuitos genéticos.

### 1.2 Objetivos

Este proyecto tiene dos objetivos principales, el primero es añadir la funcionalidad de comunicación intercelular mediante conjugación al simulador GRO, y el segundo es usar datos experimentales recibidos por un laboratorio de microbiología. Para ello, se seguirán los siguientes puntos:

- Estudio de la biología de conjugación como mecanismo de comunicación intercelular.
- Diseño e implementación del algoritmo que simula la conjugación.
- Validación experimental y ajuste del modelo a los datos experimentales sobre las tasas de conjugación y crecimiento de las bacterias

### 1.3 Plan del trabajo

Este documento detalla las modificaciones añadidas al simulador GRO para que pueda reproducir el proceso de conjugación bacteriana. Esta memoria está dividida en seis capítulos:

- Capítulo 1: Contextualización de la Biología Sintética y la Biología de Sistemas
- Capítulo 2: Descripción de los diversos simuladores utilizados en el ámbito de la Biología Sintética
- Capítulo 3: Aspectos teóricos sobre conjugación bacteriana necesarios para su posterior implementación.
- Capítulo 4: Descripción del simulador GRO, así como del motor físico que utiliza.
- Capítulo 5: Descripción de las nuevas funcionalidades añadidas al simulador GRO.
- Capítulo 6: Verificación del modelo, así como ejemplos de simulaciones y conclusiones.



# *Capítulo 2*

## *ESTADO DEL ARTE*

---

2.1 OTROS SIMULADORES.....	5
----------------------------	---

---

En el ámbito de la Biología Sintética existen diversos simuladores para la representación del comportamiento de una colonia. En este capítulo se les hará una ligera mención, así como las modificaciones que sufrieron algunos de ellos por parte de anteriores compañeros del grupo LIA.

### **2.1 Otros simuladores**

Existen dos modelos en los que los simuladores de colonias de bacterias se basan actualmente, el modelo AbM/IbM, que trata cada entidad de forma individualizada, y los modelos que analizan las dinámicas a nivel de población con ODEs/PDEs (Ordinary Differential Equations/ Partial Differential Equations). En este proyecto nos centraremos en los sistemas multiagente en el que cada bacteria actúa de forma individualizada e interactúa con las demás. Actualmente existen numerosos simuladores de colonias bacterianas basados en esta metodología, pero en el ámbito de la Biología Sintética son siete los más utilizados.

### **CellModeller**

CellModeller es un módulo de Python para el análisis multicelular 2D diseñado por la Universidad de Cambridge. Este módulo no fue creado especialmente para simular el comportamiento de bacterias, su función es la de simular el comportamiento de células en general, que abarca desde algas microscópicas hasta las células que intervienen en la formación de tejidos. En cambio, ofrece una gran variedad de herramientas para facilitar la configuración del usuario.

Recientemente, se le ha agregado un nuevo método para que permite la computación paralela mediante OpenCL [2]. La computación paralela permite conseguir una colonia con más número de bacterias en menos tiempo de simulación. Este software incluye modelos de la biofísica, la genética y la señalización intercelular. A partir de una sola célula consigue simular una colonia de aproximadamente 32.000 células individuales en 30 min. Reproduce las características principales de una colonia como el crecimiento exponencial del número de células permitiendo añadir limitaciones físicas duras o blandas. Posteriormente, se implementó CellModeller4, que permite la especificación del comportamiento de la célula a través de modelos basados en reglas y ecuaciones diferenciales.

### **BSim**

BSim es una herramienta de modelado altamente personalizable basada en agentes [3]. Se basa en un espacio 3D en el que las bacterias son definidas de forma esférica. Fue creado con la idea de que todos los aspectos del modelo deben ser actualizables, ampliados o reemplazados por versiones definidas por el usuario. Este enfoque permite el rápido desarrollo de simulaciones que capturan las características principales de un sistema, al tiempo que permite que estos sean redefinidos por datos experimentales según estén disponibles.

### **BacSim**

BacSim es un simulador multiagente en el que cada agente es en realidad una esfera que se representa en un espacio 2D como un círculo [4]. Por tanto, este simulador no admite la forma de bacilo en sus células, lo que afecta en gran medida al comportamiento de la colonia. El modelo describe propiedades bacterianas incluyendo captación de sustrato, el metabolismo, el mantenimiento, la división celular y la muerte en el nivel de célula individual. Para la difusión de sustrato, se utiliza una red de difusión de dos dimensiones. Con el fin de hacer el modelo fácilmente aplicable a diferentes tipos de bacterias, este software cuenta solo con ocho parámetros que definir y que pueden ser aleatoriamente definidos.

### **DiSCUS**

DiSCUS (Discrete Simulation of Conjugation Using Springs) fue, junto con BactoSim, de los primeros en ofrecer en sus simulaciones redes genéticas y comunicación intercelular mediante conjugación [5]. Las células se definen en forma de bacilo y vienen dadas por una longitud y un radio. El proceso de conjugación se representa mediante un muelle elástico que conecta a una donante con una receptora, este muelle está constantemente monitorizado durante un tiempo establecido de conjugación, en el que se comprueba que está constantemente conectado con ambas células. Además, se puede establecer si las transconjugadas tienen o no la capacidad de donar plásmidos.

### **BactoSim**

BactoSim es un simulador espacial implementado en Java. Fue creado específicamente por Antonio García (grupo LIA) para simular el comportamiento de bacterias en una colonia [6]. En este simulador, se define el espacio como una rejilla discreta en la que se colocan todos los agentes y interaccionan entre ellos. Además de tener todas las funcionalidades en cuanto a crecimiento e interacciones, Antonio añadió además la funcionalidad de conjugación con el propósito de estudiar este proceso in-sillico. Este simulador está siendo continuamente mejorado para validarse con los datos experimentales del IBBTEC.

### **iDynoMICS**

iDynoMICS (individual-based Dynamics of Microbial Communities Simulator) es un simulador de código abierto que permite trabajar con varios modelos computacionales de bacterias, en 2D y 3D [7]. Además, permite configurar una gran cantidad de parámetros respecto al entorno y reacciones químicas. En 2011 Jean Pierre Arteaga (grupo LIA) agregó un primer módulo con la capacidad de trabajar con bacterias en forma capsular, ya que inicialmente solo permitía el uso de bacterias en forma esférica [8]. Esto causó la necesidad de diseñar un nuevo algoritmo para los solapamientos de las bacterias. Posteriormente, Aurora Arroyo (grupo LIA) le añadió un segundo módulo que permitió la simulación de puertas AND mediante conjugación [9]. Actualmente aún está en fase de verificación, puesto que no se ha comprobado que funciona de una forma biológicamente correcta.

### **GRO**

GRO es un paquete de software de código abierto que combina un enfoque de sistemas distribuidos y computación paralela con la simulación de, en un principio, hasta unos pocos miles de bacterias que crecen en un entorno 2D [10]. La razón de conseguir solo unos pocos miles de bacterias es porque se basaba en un motor físico que requería un gran coste computacional debido a la constante creación de nuevos hilos de ejecución en memoria por cada nueva bacteria. Este motor fue reemplazado en 2013,

por Luis Enrique Muñoz y por mi misma (grupo LIA), por otro diseñado especialmente para este simulador para conseguir alcanzar colonias más extensas.

Se ha elegido el simulador GRO para este proyecto ya que es el idóneo para simular circuitos genéticos. Fue creado específicamente para el ámbito de la Biología Sintética. Permite el uso de señales, además de que integra un módulo para reproducir la expresión de proteínas de cada célula, lo cual es una característica indispensable en los circuitos genéticos. Posteriormente, en el capítulo 4, se detallarán todas las características de este simulador.

## *Capítulo 3*

### *FUNDAMENTOS*

---

3.1	PLÁSMIDOS.....	9
3.2	CONJUGACIÓN .....	10

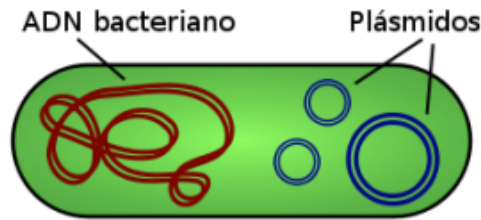
---

Este capítulo se detallan los fundamentos teóricos necesarios para poder añadir la conjugación bacteriana a GRO. Para ello, primero se definirán todas las características del material genético que se transmite en la conjugación, los plásmidos, y posteriormente se explicará detalladamente el proceso de conjugación.

#### **3.1 Plásmidos**

Las bacterias, a parte del cromosoma, contienen moléculas de ADN ajenas al cromosoma llamadas plásmidos [11][12]. Un plásmido es un fragmento de ADN circular de doble hélice que se replica y se transcribe de forma independiente al cromosoma. Este fragmento de material genético puede transmitirse de una célula a otra permaneciendo inmutable el cromosoma. En cuanto al tamaño, los plásmidos tienen tamaño variable desde menos de 10 hasta 400 pares de kilobases. Además, una bacteria no tiene por qué tener un solo plásmido, de hecho puede tener varias copias del mismo plásmido o distintos plásmidos, aunque es difícil encontrar plásmidos con cierta semejanza de ADN entre sí.





**Ilustración 1. Plásmido conjugativo extracromosómico en el interior de la célula procariota. [11]**

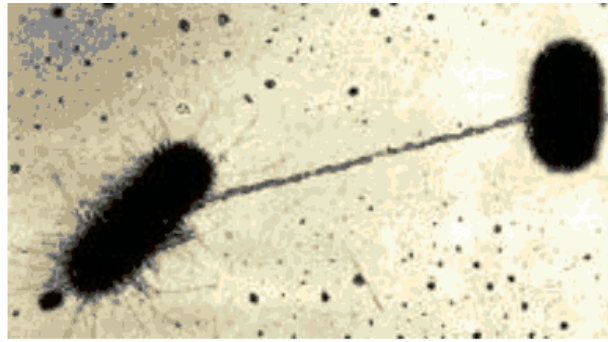
Los plásmidos, además de codificar resistencia a los antibióticos, pueden codificar factores de virulencia, lo que puede causar que la bacteria resulte patógena. El gran interés que muestra el proyecto PLASWIRES en los plásmidos proviene de la idea de que los plásmidos pueden ejercer la función de cables intercelulares para crear un sofisticado circuito multicelular. Los plásmidos pueden ser programados, por ejemplo, con puertas lógicas y así conseguir que hagan la función de antibióticos programables o de plásmidos centinela que detecten plásmidos peligrosos en bacterias virulentas. El proceso biológico que utilizan los plásmidos para ejercer esa función de cable intercelular es la conjugación bacteriana.

### 3.2 Conjugación

La conjugación bacteriana es el proceso de transferencia horizontal de genes desde una célula donante a otra receptora [11][12]. Normalmente, en la transferencia únicamente se transmite el plásmido, pero hay casos en que estos plásmidos se comportan como episomas. En este caso, se trasmite parte del cromosoma además del plásmido. Esta parte del cromosoma se adherirá en el cromosoma de la receptora y se formará en ella una combinación con el cromosoma nuevo.

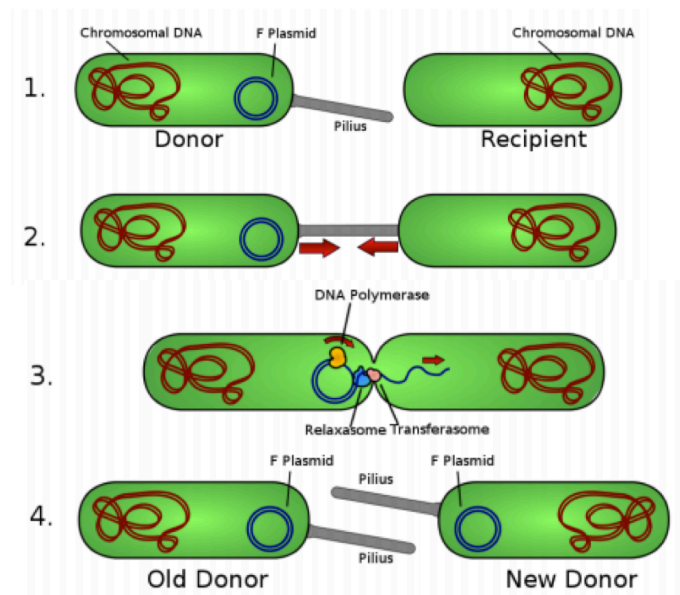
Este proceso requiere un contacto directo entre las dos bacterias que intervienen en el proceso. Toda bacteria *E. Coli* esta formada por una serie de filamentos cortos y estrechos, que son huecos y rectos llamados pilus. Estos pilus están compuestos por proteínas y su función principal es mediar la adhesión. Existen dos tipos:

- Sexuales: (1-10 por célula). Tienen como función específica la transmisión de material genético en el proceso de conjugación.
- No sexuales: (100-200 por célula). Tienen como función específica la adhesión a tejidos y superficies.



**Ilustración 2. Proceso de conjugación obtenido por microscopía electrónica [6].** Bacteria donante transfiriendo plásmido a través del pilus sexual a una bacteria receptora.

Una vez que el pilus se ha adherido a otra bacteria se produce el proceso de conjugación. Durante este proceso, una de las cadenas de ADN del plásmido de la bacteria donante se desenrolla y se transmite a través del pilus a la bacteria receptora. En el interior de la bacteria receptora se arma la primera hebra circular. Al mismo tiempo se sintetiza la segunda hebra circular regenerándose en ambas bacterias el plásmido completo.



**Ilustración 3. Proceso de conjugación [11].** Transferencia de plásmido de una bacteria donante a una receptora.

En este momento, la bacteria receptora pasa a ser transconjugada teniendo también posteriormente la capacidad de conjugar, comportándose como donante.



# Capítulo 4

## ***SIMULADOR GRO***

---

4.1	SIMULADOR GRO .....	13
4.2	MOTOR FÍSICO CELLENGINE .....	15

---

Este capítulo describe las características del simulador GRO, así como las del nuevo motor físico CellEngine, el cual ha sido creado específicamente para asumir la física de colonias bacterianas.

### **4.1 Simulador Gro**

GRO es un simulador multicelular que modela el comportamiento de bacterias en una colonia [10]. Este simulador reproduce el movimiento, el crecimiento y las divisiones de las células. Además, permite el uso de señales que afectan al comportamiento de ciertas sustancias químicas sobre las bacterias.

Las células se definen inicialmente con un diámetro de  $1\ \mu m$  y un largo de  $2\ \mu m$  lo que resulta un volumen de  $1.57\ fL$  [3]. Normalmente, las bacterias crecen  $0.034\ fL/min$ , por eso ese es el valor por defecto que utiliza GRO para el crecimiento, pero este se puede definir al principio de la simulación.

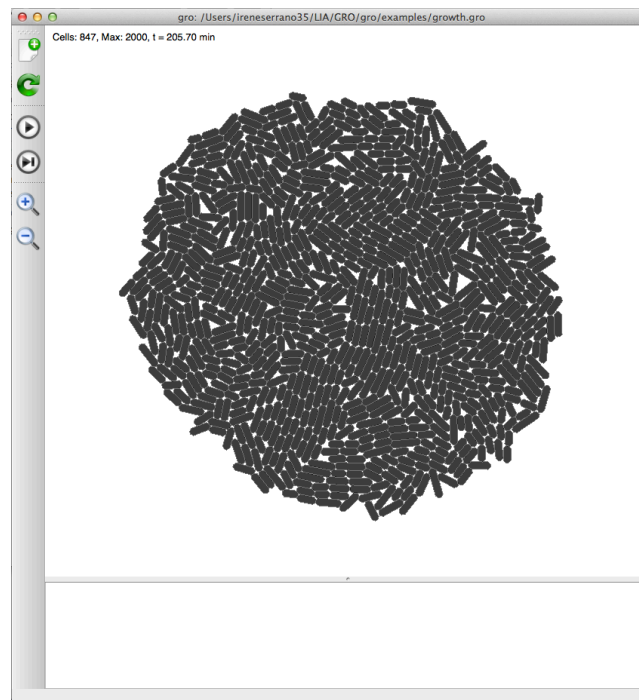
GRO resume la variación de volumen en la fórmula  $V' = V + \delta kV$  donde  $\delta$  es aproximadamente la variación de tiempo, que de aquí en adelante denominaremos  $dt$ . Este  $dt$  es también la duración de una iteración en la simulación, por eso, si se define

una variación de tiempo muy alta se conseguirán simulaciones menos precisas, pero si en cambio se define un  $dt$  muy bajo el crecimiento será mas lento y habrá que esperar mucho tiempo para conseguir el resultado esperado.

Cada bacteria crece hasta que aproximadamente dobla su tamaño, en este punto se divide. El volumen en el que las bacterias se dividen por defecto es de  $3.14 fL$  con una varianza de  $0.005 fL^2$ . Ambos parámetros pueden ser definidos antes de comenzar una simulación.

El hecho de que se puedan definir ciertos parámetros es debido a que este simulador tiene su propio lenguaje de programación denominado igual que él, *gro*. Se trata de un lenguaje funcional bastante sencillo basado en un conjunto de reglas que posteriormente el interpreta. Este lenguaje permite definir todos los parámetros que serán dados a la colonia para su posterior simulación. A continuación se muestra un ejemplo del lenguaje en el que se define la variación de tiempo  $dt$  a 0.1 y la posición de una nueva bacteria E. Coli, que en este caso la posiciona en el punto (0,0). Una vez ejecutado este ejemplo, solo hay que esperar unos minutos para que GRO consiga crear una colonia de 847 bacterias.

```
include gro
set ( "dt", 0.1 ); // fast and inaccurate
program p() := {
  skip();
};
ecoli ( [ x := 0, y := 0 ], program p() );
```



**Ilustración 4: Crecimiento de una colonia bacteriana en GRO.** Simulación realizada previamente a las nuevas funcionalidades añadidas.

En conclusión, se ha considerado que GRO es la mejor opción entre los simuladores disponibles para añadirle la funcionalidad de conjugación, ya que es un software de código abierto y como se ha descrito anteriormente permite definir numerosos parámetros que son importantes para poder asemejarse lo máximo posible a la realidad.

## 4.2 Motor Físico CellEngine

CellEngine es un motor físico que actualmente es la base de todos los procesos físicos que utiliza GRO. En concreto, su función es la de calcular todas las fuerzas que ejercen unas bacterias contra otras y asegurarse de que no hay solapamientos de unas con otras.

En un principio, GRO utilizaba el motor físico *Chipmunk*, pero éste ralentizaba el simulador al no ser específico para bacterias. Este motor representaba las bacterias como un polígono de ocho vértices, que para comprobar si existe solapamiento entre dos bacterias, tenía que comprobar si cada uno de esos ocho vértices estaba posicionado en el interior del polígono que representa otra bacteria. Este problema causaba que GRO no consiguiera alcanzar poblaciones de más de 1000 bacterias, por tanto, el semestre pasado mi compañero Luis Enrique Muñoz y yo realizamos un nuevo motor físico, *CellEngine*. Este nuevo motor físico se implementó especialmente para albergar toda la física de una colonia de bacterias y conseguir que GRO alcance poblaciones más numerosas.

En CellEngine, cada célula viene definida básicamente por tres puntos, los puntos P1, P2 y el punto central. El lugar geométrico del segmento que une estos tres puntos es lo que da la forma de capsula. Además, cuenta con parámetros como el radio y el ángulo  $\theta$  que tiene la bacteria respecto del eje horizontal. El vector unitario y el ortogonal se definen en la célula para agilizar los cálculos.

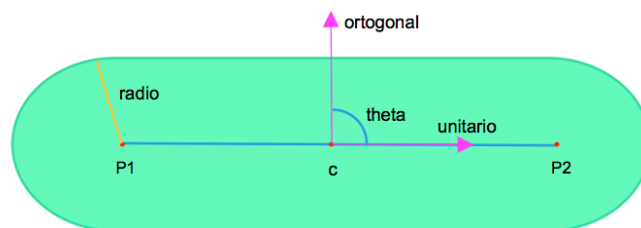


Ilustración 5. Definición de una célula en CellEngine.

CellEngine utiliza un nuevo algoritmo basado en anillos en el que actualmente está trabajando Luis Enrique Muñoz que se encarga de tratar la física por anillos desde el centro hacia el exterior de la colonia y así minimizar costes de computación, ya que solo realiza la física con las bacterias de su anillo.

La función principal del motor es realizar la relajación de las bacterias, es decir, asegurarse de que no hay solapamientos entre ellas. Para comprobar que existe solapamiento, lo primero es crear dos triángulos partiendo de  $P_1$  y  $P_2$ , uno a  $P_1$  y el otro a  $P_2$  de la bacteria contigua, quedando los triángulos  $\overline{P_1P_2P_{1c}}$  y  $\overline{P_1P_2P_{2c}}$ . Lo siguiente es hallar las alturas de estos dos triángulos. Si alguna de esas alturas es menor que dos veces el radio y resulta dentro del segmento  $\overline{P_1P_2}$ , significa que hay solapamiento. En la ilustración 6 pueden verse algunos ejemplos de este algoritmo. Una vez comprobado, el motor procederá a hallar las fuerzas que ejercen una contra otra y relajarlas para que no solapen.

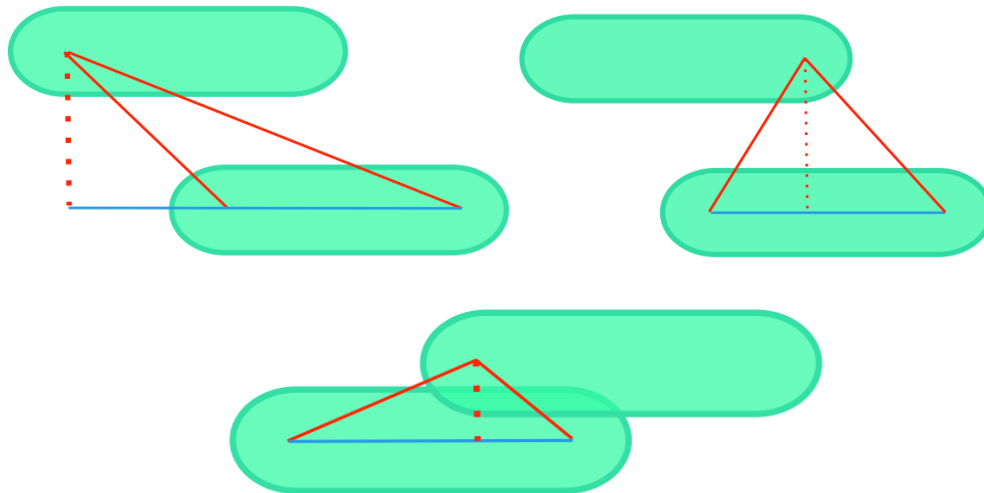


Ilustración 6. Cálculo de solapamientos

# Capítulo 5

## NUEVAS FUNCIONALIDADES

---

5.1	DONANTES, RECEPTORAS Y TRANSCONJUGADAS.....	17
5.1.1	Datos experimentales.....	18
5.2	CONJUGACIÓN.....	20

---

En este capítulo se detallarán las nuevas funcionalidades que han sido añadidas al simulador GRO. La principal funcionalidad es la comunicación intercelular mediante conjugación, que permite visualizar como las bacterias se transmiten plásmidos. Además, se han ajustado algunas dinámicas con los datos experimentales proporcionados por el Instituto de Biomedicina y Biotecnología de Cantabria (IBBTEC)[13].

### 5.1 Donantes, Receptoras y Transconjugadas

El proceso de conjugación requiere diferenciar las bacterias E. Coli en tres estados: Donante, Transconjugada y Receptora. Como su nombre indica, las bacterias donantes serán las portadoras de un plásmido. Las transconjugadas serán las bacterias que acaban de recibir un plásmido pero que aun no están estabilizadas para comportarse completamente como donantes, aunque tengan igualmente la capacidad de conjuguar. Y por último las receptoras, que recibirán el plásmido.



### 5.1.1 Datos experimentales

Según los últimos estudios del IBBTEC, la bacteria E. Coli tiene comportamientos distintos dependiendo del estado en el que se encuentre [13]. Por esta razón, es necesario conocer todas estas diferencias para conseguir ajustar el modelo para que sea biológicamente correcto.

El primer parámetro a considerar es el *Tiempo de Generación (Generation Time)*, el cual define el tiempo que tarda una bacteria en dividirse. Este parámetro viene dado por un tiempo y una varianza. E. Coli tiene un tiempo de Generación diferente para cada estado, y la siguiente tabla resume los valores.

	Tiempo	Varianza
Receptora	23.16	1.0
Donante	28.7	2.1
Transconjugada	27.99	2.0

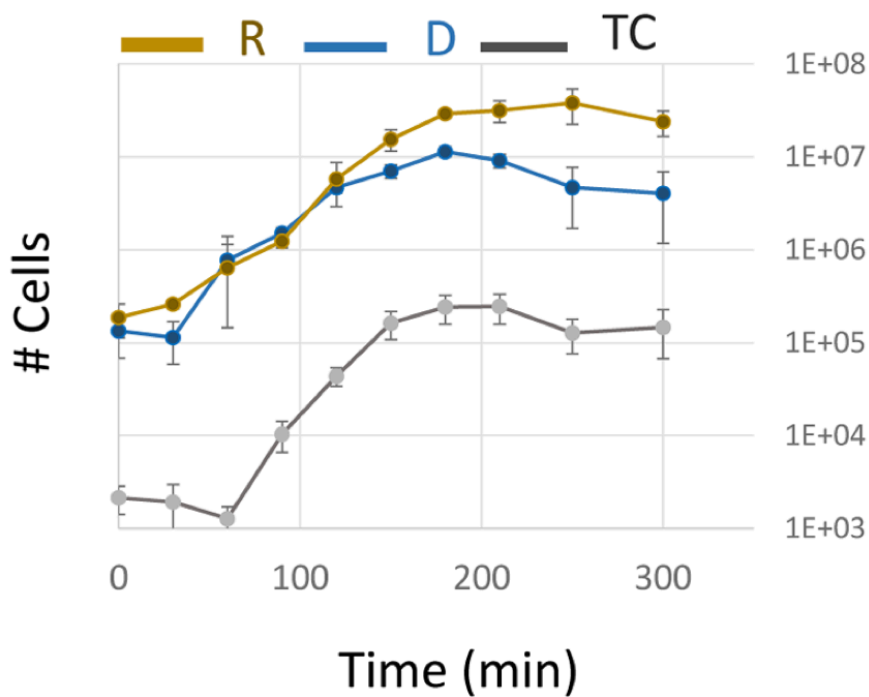
**Tabla 1. Tiempo de Generación[13].** Se representan los tiempos medios y la varianza estándar de 12 experimentos.

Con estos datos se ajustan las tasas de crecimiento en el simulador para que sean proporcionales al tiempo de generación. En concreto, dividiremos 1.57, que es el volumen que aumenta en promedio cada bacteria, entre el tiempo de Generación para conseguir la tasa de crecimiento. La siguiente tabla tiene valores orientativos de cómo serían las tasas de crecimiento de una bacteria por minuto, ya que los valores se han hallado sin suponer la varianza.

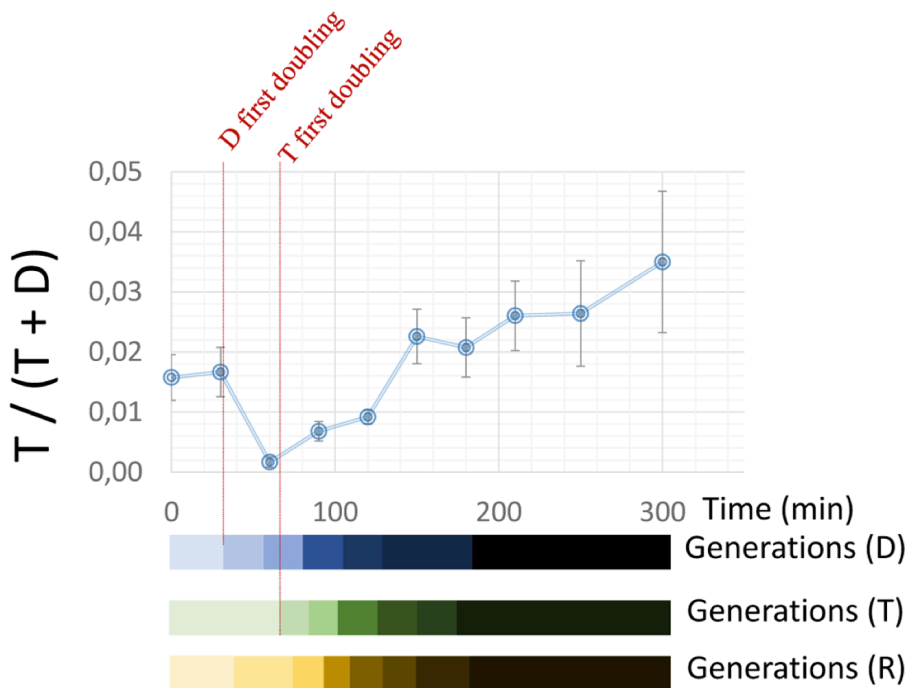
	Tasa
Receptora	0.067789
Donante	0.054703
Transconjugada	0.056091

**Tabla 2. Tasa de Crecimiento.** Calculadas a partir del tiempo del tiempo de Generación.

Como se puede ver en la tabla, cuando una bacteria es receptora crece más rápido que en el resto de estados. Los estados de transconjugada y donante tienen parecidas tasas de crecimiento pero a las transconjugadas les ocurre una curiosa alteración al recibir el plásmido.



**Ilustración 7. Número de bacterias respecto del tiempo [13].** Datos tomados por el IBBTEC partiendo de una relación 1:1 de Donantes/Receptoras y 1:1000 de Transconjugadas/Donantes.



**Ilustración 8. Relación T/(T+D) [13].** Datos tomados por el IBBTEC partiendo de una relación 1:1 de Donantes/Receptoras y 1:1000 de Transconjugadas/Donantes.

En estudios recientes del IBBTEC se ha observado que las transconjugadas que acaban de recibir un plásmido tardan en volverse a dividir alrededor de 3 veces mas que el resto de células. En cambio, una vez divididas por primera vez, vuelven a recobrar las tasas originales e incluso un poco mas altas ya que al final consiguen llegar al mismo número de divisiones en tiempo que las demás [13]. La ilustración 7 demuestra la afirmación anterior, ya que el número de transconjugadas disminuye notablemente, señal de que no han habido divisiones.

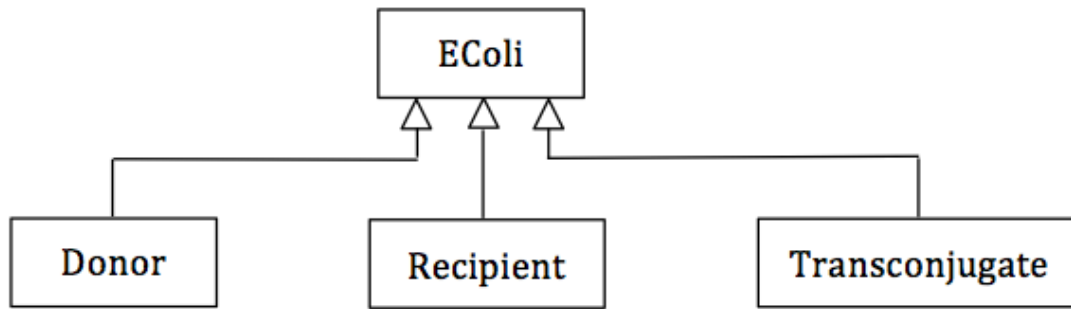
Por tanto, la tasa de crecimiento de las transconjugadas decrece hasta que se dividen por primera vez. Este suceso también puede demostrarse en la ilustración 8, donde puede verse la relación  $T/(T + D)$ , donde T es el número de transconjugadas y D el número de donantes. En este gráfico se aprecia como esta relación sufre una decaída justo antes de que ocurra la primera división de una transconjugada. Esto ocurre porque las donantes ya se han dividido y su número se ha duplicado mientras que el número de transconjugadas sigue siendo el mismo ya que siguen creciendo a una velocidad muy baja.

Otro proceso interesante que sufren las transconjugadas es el denominado *overshooting*. Durante las cinco primeras generaciones de una transconjugada, ésta sufre un aumento en la probabilidad de conjugación, haciendo que conjugue más que las bacterias donantes. Este proceso no está implementado en la actualidad pero es importante su mención para la posterior interpretación de los datos.

## 5.2 Conjugación

El proceso de conjugación consiste en el traspaso de un plásmido de una bacteria a otra [14]. En el momento en que una bacteria receptora recibe un plásmido, ésta se convierte en transconjugada, que posteriormente, tras 3 generaciones se estabilizará y se convertirá en donante. Durante el tiempo que pasa mientras que la transconjugada se convierte en donante, ésta también tiene la capacidad de conjuguar, por lo que es necesario añadir el método de conjugación, igualmente a las bacterias transconjugadas.

El primer paso es crear la diferenciación de estados de una bacteria E. Coli. GRO contiene la clase EColi, la cual se encarga del crecimiento y la división de cada una de las células. Para crear los nuevos estados, la mejor opción consiste en crear tres nuevas clases Donor, Recipient y Transconjugate como clases que heredan de EColi.



**Ilustración 9. Herencia EColi.** Esquema que ilustra la herencia utilizada en la creación de las nuevas tres clases (Donor, Recipient, Transconjugate).

Se ha decidido crear los métodos de crecimiento y división implícitos de cada uno de los tres tipos por simplificación. Esta no es una medida totalmente necesaria ya que todas las células crecen y se dividen por igual, sólo diferenciándose en las tasas, pero dado que tienen diferentes tasas de crecimiento y que las bacterias transconjugadas sufren una ralentización, se ha decidido crear un método distinto en cada una de las tres clases con sus tasas correspondientes.

Una vez creadas las tres clases y comprobado que el crecimiento y la división siguen funcionando correctamente, llega el momento de implementar la conjugación. Para ello, es necesario crear un método que decida cuando una bacteria puede conjugarse.

$\gamma_0$  es la tasa de conjugación. Esta depende del número de bacterias que una célula tiene alrededor y el número de conjugaciones que realiza en su ciclo de vida, es decir, antes de dividirse. Por tanto, se tiene la siguiente relación.

$$\gamma_0 = \frac{c}{e} \quad \begin{array}{l} c \equiv n^{\circ} \text{ conjugaciones} \\ e \equiv n^{\circ} \text{ de encuentros} \end{array}$$

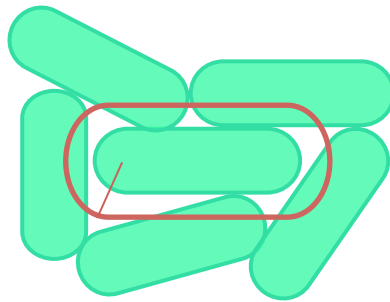
Saber qué vecinas tiene una bacteria tiene un coste computacional que supondría una gran carga para el simulador en caso de tener que calcular las vecinas de cada bacteria en cada iteración. Por esta razón, no podemos saber cuántas vecinas tiene una bacteria a priori. La solución a este problema consiste en definir externamente el valor de la tasa  $\gamma_0$ .

$\gamma_0$  define el número de conjugaciones en el ciclo de vida de una bacteria, por lo tanto hay que tener en cuenta el número de iteraciones que hará en su ciclo de vida. Este número de iteraciones se halla con el cociente entre el tiempo de Generación y el dt del programa, es decir,  $(G/dt)$ . Por tanto lo único que quedaría sería dividir  $\gamma_0$  por este cociente para conseguir la probabilidad de conjugación en cada iteración.

La probabilidad de que una bacteria conjuque es la misma que la probabilidad de que un número aleatorio salga menor que la probabilidad anteriormente descrita. Por lo tanto para que una bacteria conjuque se tiene que cumplir la siguiente condición:

$$r \leq \frac{\gamma_0}{\left(\frac{G}{dt}\right)} \text{ con } r \in [0,1]$$

La conjugación se basa en conocer las bacterias que rodean a la bacteria que contiene el plásmido. Para ello se ha recurrido al motor físico y se ha creado un nuevo método que halle una lista de vecinas. Este método realiza la misma metodología el que proceso de hallar los contactos pero con un aura de la bacteria, es decir, se duplica el radio de la bacteria y se procede a comprobar con qué bacterias colisiona. Esas bacterias formarán la lista de vecinas que se usará para conjuagar.



**Ilustración 10. Aura para el cálculo de bacterias vecinas.**  
Representación de la metodología utilizada en el cálculo de las bacterias vecinas.

Una vez hallada la lista de vecinas que rodean a la bacteria, se pasa al proceso de conjuagar en sí. Lo primero es elegir aleatoriamente a una de las vecinas de la lista y comprobar que es una Receptora para poder conjuagarle un plásmido. A continuación, se crea una nueva Transconjugada asociándole el mismo cuerpo del motor físico y volumen que la Receptora. Se elimina la Receptora de la colonia y se añade la Transconjugada. Por último se cambia el indicador de conjugación de la Donante a falso para que no vuelva a conjuagar en la siguiente iteración.

Debido a que la bacteria vecina elegida puede no ser Receptora se han decidido crear dos versiones del simulador. Una versión simplificada en la que si la vecina es Donante o Transconjugada no cuenta como conjugación válida y por tanto volverá a intentar conjuagar con otra bacteria en la siguiente iteración. Y una versión completa en la que si, igualmente, la vecina es Donante o Transconjugada cuenta como intento de conjugación y por tanto pierde la oportunidad de conjuagar.

# Capítulo 6

## RESULTADOS Y CONCLUSIONES

---

6.1	SIMULACIONES.....	23
6.1.1	<i>Versión Simplificada</i> .....	24
6.1.2	<i>Versión Completa</i> .....	26
6.2	RESULTADOS.....	28
6.2.1	<i>Versión Simplificada</i> .....	28
6.2.2	<i>Versión Completa</i> .....	31
6.3	CONCLUSIONES.....	34
6.4	TRABAJOS FUTUROS.....	35

---

En este capítulo se mostrarán los resultados obtenidos al introducir la comunicación intercelular al simulador GRO. Además, en el apartado *Resultados* se verificará que se consigue simular este proceso de forma muy aproximada a como ocurre biológicamente.

### 6.1 Simulaciones

A continuación se muestran ilustraciones que representan el funcionamiento del simulador con ambas versiones. En ambos ejemplos se comienza con 50 bacterias receptoras y 5 donantes, todas ellas repartidas aleatoriamente.

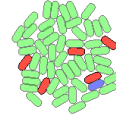
En el simulador se han representado las Donantes en color rojo, las Transconjugadas en color azul y las Receptoras en color verde.

### 6.1.1 Versión Simplificada

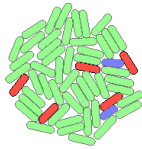
Cells: 55, Donors: 5, Transconjugates: 0, Recipients: 50,  
Gamma0: 0.14286, Max: 20000,  $t = 0.00$  min



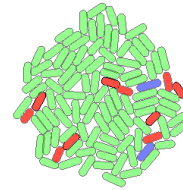
Cells: 55, Donors: 5, Transconjugates: 1, Recipients: 49,  
Gamma0: 0.14286, Max: 20000,  $t = 5.05$  min



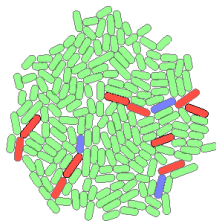
Cells: 55, Donors: 5, Transconjugates: 2, Recipients: 48,  
Gamma0: 0.14286, Max: 20000,  $t = 20.25$  min



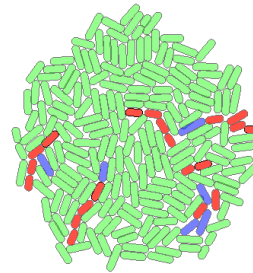
Cells: 108, Donors: 10, Transconjugates: 2, Recipients: 96,  
Gamma0: 0.14286, Max: 20000,  $t = 35.05$  min



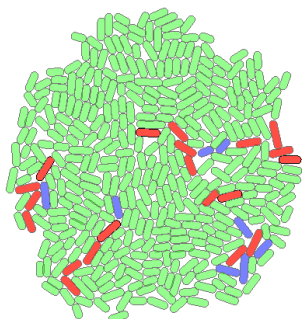
Cells: 196, Donors: 10, Transconjugates: 3, Recipients: 183,  
Gamma0: 0.14286, Max: 20000,  $t = 50.25$  min



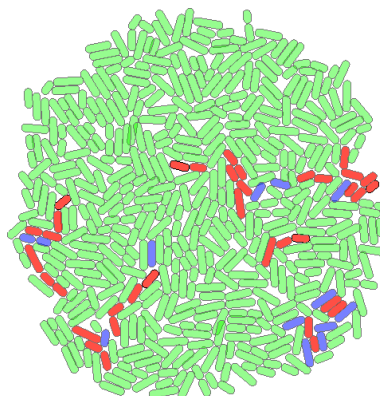
Cells: 219, Donors: 20, Transconjugates: 6, Recipients: 193,  
Gamma0: 0.14286, Max: 20000,  $t = 62.25$  min



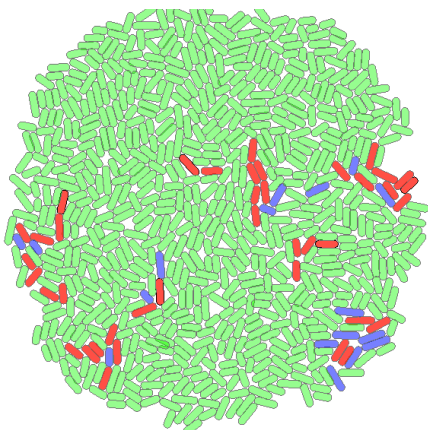
Cells: 397, Donors: 20, Transconjugates: 8, Recipients: 369,  
Gamma0: 0.14286, Max: 20000, t= 75.05 min



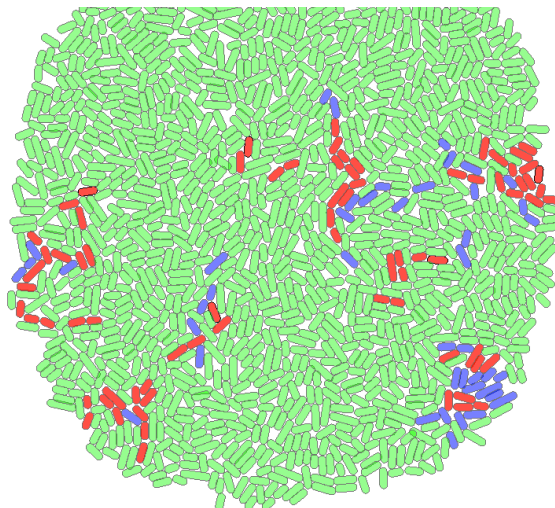
Cells: 520, Donors: 36, Transconjugates: 13, Recipients: 471,  
Gamma0: 0.14286, Max: 20000, t= 90.25 min



Cells: 787, Donors: 40, Transconjugates: 17, Recipients: 730,  
Gamma0: 0.14286, Max: 20000, t= 101.35 min



Cells: 1440, Donors: 76, Transconjugates: 44, Recipients: 1320,  
Gamma0: 0.14286, Max: 20000, t= 120.20 min



**Ilustración 11. Simulación versión simplificada.** Ejemplo de una ejecución en la versión simplificada del simulador GRO partiendo de 5 donantes y 50 receptoras.

En esta simulación se puede apreciar como la bacteria transconjugada que se crea al inicio sufre la ralentización propia de su estado ya que todas las receptoras y donantes de su alrededor llegan a dividirse dos veces mientras que esa bacteria sigue creciendo lentamente sin haberse dividido aun.

En la última imagen se aprecia que se ha llegado a 44 transconjugadas en algo mas de 120 minutos, este número es bastante alto ya que es la versión simplificada, y como ya se ha comentado antes, si la bacteria que elige para congregar no es receptora vuelve a intentarlo en la siguiente iteración con otra vecina.

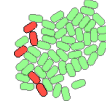


### 6.1.2 Versión Completa

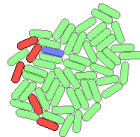
Cells: 55, Donors: 5, Transconjugates: 0, Recipients: 50,  
Gamma0: 0.14286, Max: 20000, t= 0.00 min



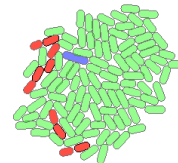
Cells: 55, Donors: 5, Transconjugates: 0, Recipients: 50,  
Gamma0: 0.14286, Max: 20000, t= 0.35 min



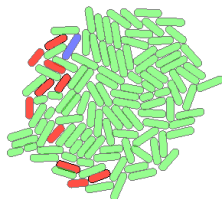
Cells: 55, Donors: 5, Transconjugates: 1, Recipients: 49,  
Gamma0: 0.14286, Max: 20000, t= 13.80 min



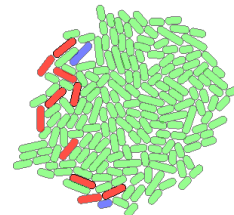
Cells: 109, Donors: 10, Transconjugates: 1, Recipients: 98,  
Gamma0: 0.14286, Max: 20000, t= 29.50 min



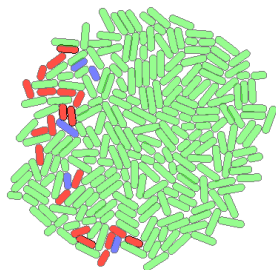
Cells: 116, Donors: 10, Transconjugates: 1, Recipients: 105,  
Gamma0: 0.14286, Max: 20000, t= 42.40 min



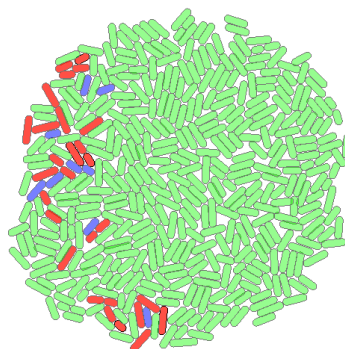
Cells: 189, Donors: 10, Transconjugates: 2, Recipients: 177,  
Gamma0: 0.14286, Max: 20000, t= 48.75 min



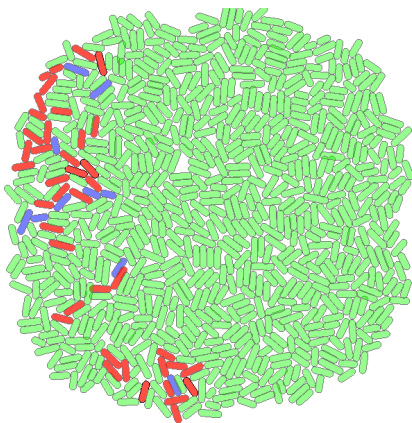
Cells: 247, Donors: 20, Transconjugates: 5, Recipients: 222,  
Gamma0: 0.14286, Max: 20000, t= 65.85 min



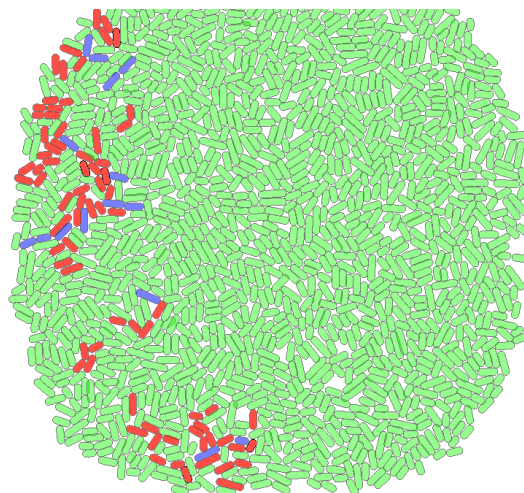
Cells: 434, Donors: 28, Transconjugates: 9, Recipients: 397,  
Gamma0: 0.14286, Max: 20000, t= 85.00 min



Cells: 826, Donors: 41, Transconjugates: 10, Recipients: 775,  
Gamma0: 0.14286, Max: 20000, t= 105.20 min



Cells: 1481, Donors: 76, Transconjugates: 15, Recipients: 1390,  
Gamma0: 0.14286, Max: 20000, t= 120.10 min



**Ilustración 12. Simulación versión completa.** Ejemplo de una ejecución en la versión completa del simulador GRO partiendo de 5 donantes y 50 receptoras.

Al igual que en la versión simplificada, se aprecia como la bacteria transconjugada rodeada en amarillo no se divide hasta la séptima imagen.

En este caso, se consiguen 15 transconjugadas en la última imagen, un número bastante mas bajo que en la versión simplificada. Razonablemente, en esta versión se han de conseguir números más bajos de conjugaciones ya que en caso de elegir una bacteria que no es receptora sigue contando como conjugación productiva.

## 6.2 Resultados

Se han realizado 10 simulaciones de cada versión para compararlas con los datos obtenidos del IBBTEC. Ya que en los datos recibidos se parte del mismo número de receptoras que de donantes, en estas simulaciones se partirá de 25 bacterias receptoras y 25 donantes repartidas aleatoriamente y con distintos tamaños iniciales. Cada 5 minutos de simulación se recogerán datos sobre el número de bacterias de cada estado y se hallará la relación  $T/(T + D)$ .

### 6.2.1 Versión Simplificada

La siguiente tabla recoge el promedio de los datos en las 10 simulaciones obtenidas en la versión simplificada. Con estos datos se han obtenido los gráficos de las ilustraciones 13 y 14, uno para el número de bacterias respecto del tiempo y otro para los valores de  $T/(T + D)$ .

Tiempo(min)	Donantes	Receptoras	Transconj.	$T/(T + D)$	Desv. Est. $T/(T + D)$
5	25	24,5	0,5	0,018945869	0,026531698
10	28,8	33,1	2	0,063759297	0,045279491
15	37,7	32,4	3,1	0,075380864	0,043308191
20	37,7	32	3,7	0,088707622	0,04656316
25	37,7	42,2	5,2	0,119195407	0,054951201
30	48,6	45,9	6,7	0,119446942	0,042005708
35	51,5	57,8	8,2	0,134501309	0,048955501
40	68	58,6	8,6	0,110987852	0,042206468
45	75,3	64,9	9,9	0,114839022	0,038730094
50	75,4	80,3	11,5	0,130577546	0,040129971
55	80,4	94,1	14,1	0,146384631	0,048377247
60	96,5	107,7	17,1	0,148752698	0,04047846
65	112,9	110,4	19,6	0,146326465	0,03720281
70	139,2	129,4	22,2	0,136870422	0,031016677
75	150	153,8	25,7	0,145611172	0,028585095
80	152,4	184	30,4	0,165168298	0,031949001
85	169	199,6	34,7	0,169421698	0,026753447
90	199,1	213,3	39,4	0,164476839	0,022846219
95	240,6	253,5	47,1	0,163629741	0,020848008
100	283,2	300,6	52,5	0,156473587	0,017023788
105	300,4	349,9	61,9	0,170546482	0,021736899
110	311,8	377	70,1	0,183096227	0,020114007
115	354,2	417	80,8	0,185291793	0,016201701
120	413,4	494,9	96,6	0,188983774	0,015556529

125	498	583,2	110,5	0,18157163	0,016744377
130	573,5	668	126,2	0,180550177	0,015929908
135	604,5	718,2	143,9	0,192457516	0,015423148
140	641,8	815,5	163,1	0,20260766	0,015769282
145	737,1	968	178,9	0,194933218	0,014530516
150	878,3	1132,2	195,4	0,181720054	0,014781997
155	1050,9	1269	216,5	0,170667293	0,012258679
160	1182,7	1385,3	244,7	0,171259884	0,013796006
165	1268,5	1590,2	269,8	0,17529459	0,016462513
170	1370,3	1870	298,6	0,178894857	0,017512521
175	1563,2	2167,4	334	0,17593196	0,018182405
180	1828,3	2412,1	374,6	0,170016162	0,020377987
185	2138,6	2654,6	428	0,166743991	0,019269832
190	2405,2	3035,6	479,8	0,166419649	0,01949209
195	2596,6	3533,5	535,7	0,171133043	0,021441358
200	2802,9	4041,6	593,7	0,174792993	0,021756351

Tabla 3. Datos versión simplificada

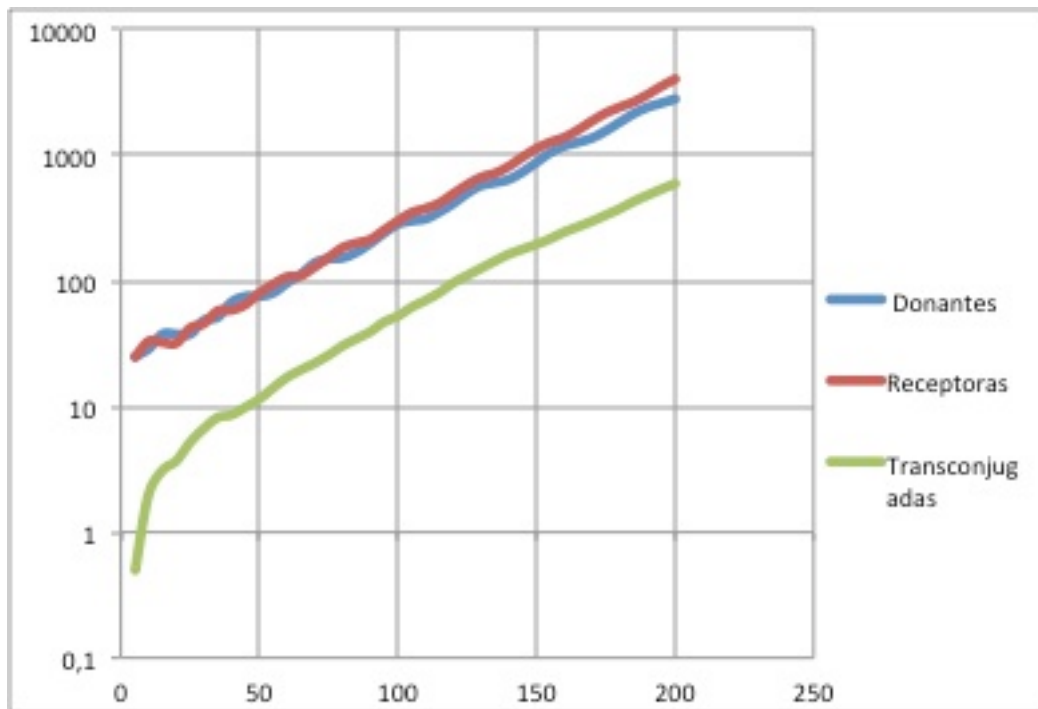
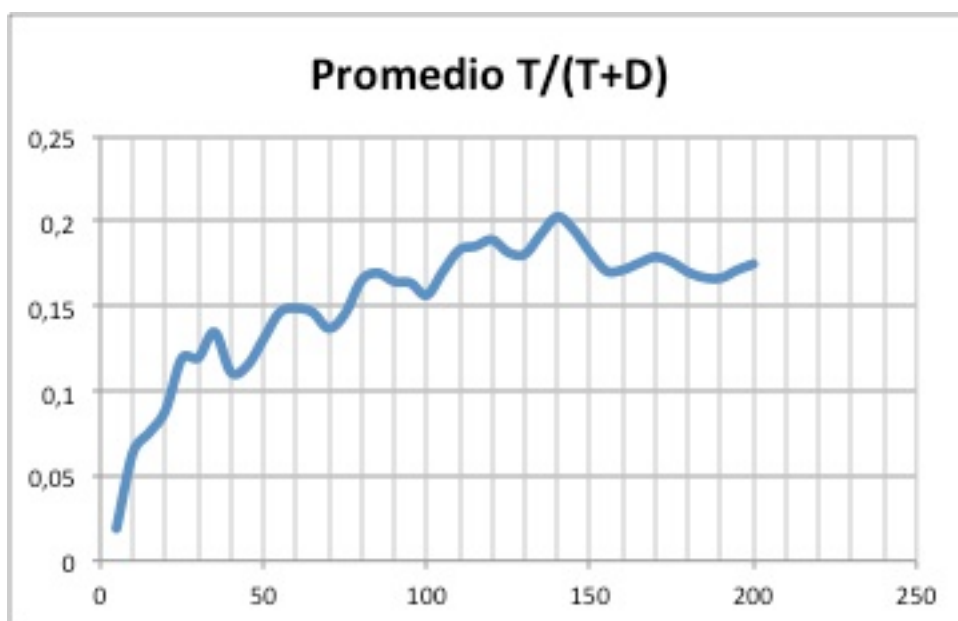


Ilustración 13. Número de bacterias versión simplificada. Representación del número de bacterias respecto al tiempo en la versión simplificada del simulador GRO. Se parte de 25 bacterias donantes y 25 receptoras.



**Ilustración 14.  $T/(T+D)$  versión simplificada.** Representación de la relación  $T/(T+D)$  respecto al tiempo en la versión simplificada del simulador GRO. Se parte de 25 bacterias donantes y 25 receptoras.

En la ilustración 13, el número de donantes y receptoras siguen la misma tendencia exponencial que los datos del IBBTEC (Ilustración 7) hasta el minuto 200, que es el punto hasta donde llega esta simulación. Ambos tipos se mantienen cercanos en número hasta el minuto 100, donde las receptoras aumentan en número dado que su tasa de crecimiento es mayor que la de las donantes.

En cuanto a las transconjugadas, primero hay que tener en cuenta que en el IBBTEC parten de una relación 1:1000 de transconjugadas/donantes, y dado que en estas simulaciones se comienza con 25 donantes, no se ha añadido ningún número inicial de transconjugadas. Dicho esto, puede apreciarse una subida más pronunciada al principio, al igual que en la ilustración 7, que luego mantiene un aumento constante.

En cuanto al gráfico de la ilustración 14 puede verse que sigue la misma tendencia que en el gráfico de la ilustración 8, incluso consiguiendo valores muy parecidos. Aún sin tener implementado el proceso de *overshooting*, se han conseguido datos tan aproximados debido a que en esta versión se consigue un mayor número de conjugaciones debido a que una donante persiste en el intento de conjugarse hasta que lo consigue con una receptora. Esto hace que se consigan valores mas parecidos a los que se conseguirían con el proceso *overshooting* que en la versión completa.

### 6.2.2 Versión Completa

Al igual que en la versión simplificada, en la siguiente tabla se recoge el promedio de los datos obtenidos en las simulaciones con la versión completa. Los gráficos de las ilustraciones 15 y 16 recogen el número de bacterias frente al tiempo y la relación  $T/(T + D)$  respectivamente.

Tiempo(min)	Donantes	Receptoras	Transconj.	$T/(T + D)$	Desv. Est. $T/(T + D)$
5	25	24,6	0,4	0,015384615	0,019861453
10	28,2	35,7	1,3	0,042977581	0,03059059
15	37,9	36,1	1,4	0,034392387	0,025217437
20	37,9	35,9	1,8	0,043423141	0,032641405
25	37,9	46,7	2,2	0,053044286	0,035072674
30	49,2	52,6	3	0,056208253	0,038233468
35	51,4	68,2	4,3	0,075992218	0,034635521
40	66	68,8	4,9	0,06783361	0,024066867
45	75,8	73,5	5,2	0,063291503	0,022888841
50	75,9	90,4	6,1	0,073715351	0,020891403
55	79,8	112	7,8	0,088431174	0,024789913
60	97,8	130,7	9,2	0,085317827	0,02782861
65	111,9	133,5	10,7	0,086732965	0,02431762
70	137,1	154,7	12,3	0,081890832	0,015974707
75	150,8	186,1	13,8	0,083465643	0,019542259
80	154,6	233,5	16,4	0,095656373	0,01811939
85	170,1	254,4	18,8	0,099299847	0,016184146
90	200,7	268,4	21,5	0,096478937	0,018467002
95	237,5	318,2	24,4	0,092950358	0,012347488
100	282	392,3	28,1	0,090502862	0,011656842
105	302,8	466,7	33	0,098111712	0,013755313
110	318,2	497,9	39,2	0,109432347	0,018191698
115	357,7	552,1	44,6	0,110683096	0,019239087
120	416	658,4	53,8	0,114324965	0,014774326
125	497,4	807,1	62	0,110593943	0,014764425
130	573,6	928,1	70,7	0,109418829	0,015343479
135	610,3	990,3	82,4	0,118664043	0,016856688
140	654,7	1130,9	93,1	0,124280074	0,01953248
145	752,3	1361,8	101,8	0,119013593	0,020141402
150	880,3	1645,1	110,6	0,111430642	0,017919381
155	1040,9	1838,5	124,4	0,106617663	0,015773103
160	1175,1	2003,2	140,6	0,106884801	0,015262109
165	1255,4	2329,2	161,8	0,114145233	0,015841137

170	1373,2	2801	177,9	0,114608008	0,017133439
175	1575,6	3300,2	197,3	0,111152222	0,017220091
180	1840,1	3651,5	222,8	0,107904789	0,017252628
185	2134,2	4051	253,2	0,105970425	0,017625208
190	2401,3	4712,4	285,5	0,10620067	0,017746686
195	2586,4	5616,4	320,3	0,110156955	0,016973172
200	2830,8	6504,7	357,8	0,112097476	0,016827407

Tabla 4. Datos versión completa

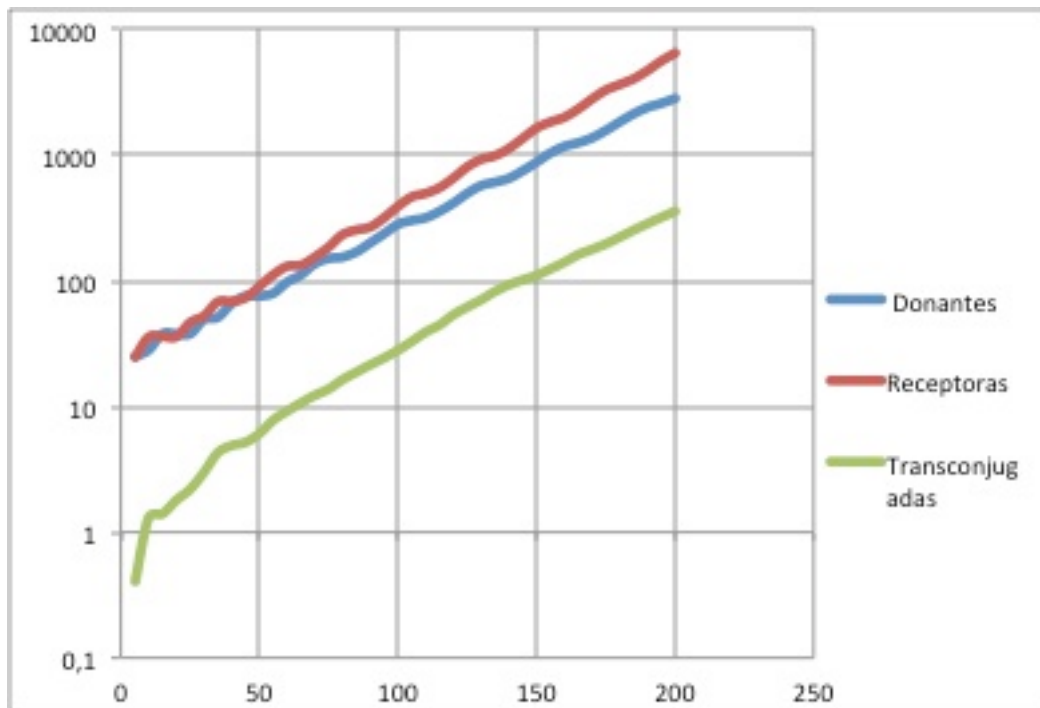
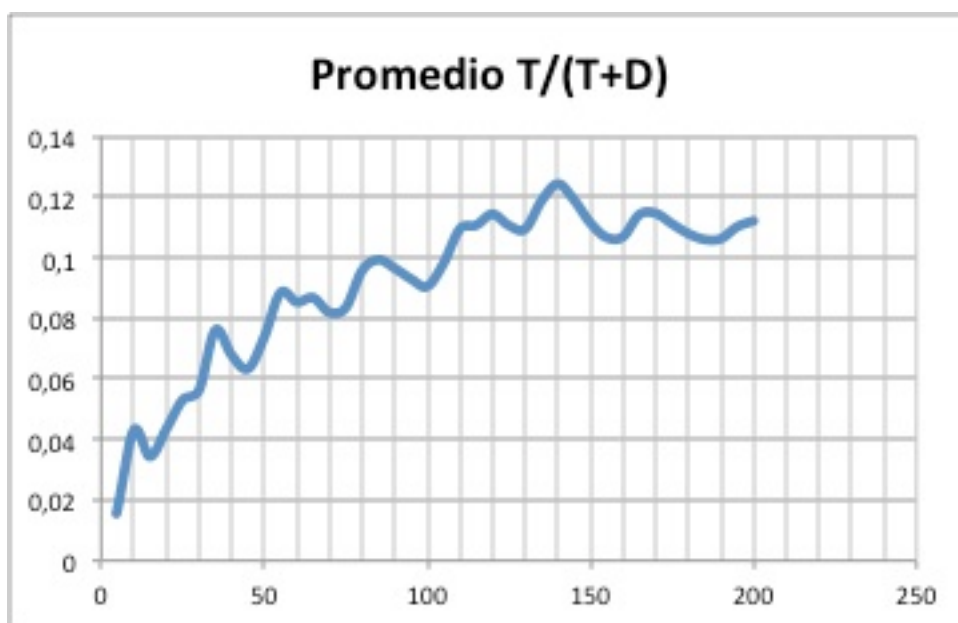


Ilustración 15 Número de bacterias versión completa. Representación del número de bacterias respecto al tiempo en la versión completa del simulador GRO. Se parte de 25 bacterias donantes y 25 receptoras.



**Ilustración 16. T/(T+D) versión completa.** Representación de la relación  $T/(T+D)$  respecto al tiempo en la versión completa del simulador GRO. Se parte de 25 bacterias donantes y 25 receptoras.

En la ilustración 15 hay mas disparidad entre el número de donantes y receptoras que en la versión simplificada, esto es debido a que en esta versión se producen menos conjugaciones dado que si una donante intenta conjugarse con otra bacteria que no es receptora cuenta como intento de conjugación y no lo vuelve a intentar. Esto hace que en esta versión no haya tantas receptoras que se conviertan en transconjugadas y por tanto, dado que la tasa de crecimiento de las receptoras es mayor, el número de receptoras aumenta más rápido que el de donantes.

En cuanto a las transconjugadas, puede apreciarse una subida menos pronunciada en los primeros 50 minutos que en la versión simplificada debido al menor número de conjugaciones. Luego, al igual que en la ilustración 7 y en la versión simplificada, mantiene un aumento constante.

En cuanto al gráfico de la ilustración 16, se observa que los datos siguen la misma tendencia pero son a la vez más bajos que en la versión simplificada y que los datos del IBBTEC. Por esta razón, hay que acudir a la definición de *overshooting*. Este proceso es el causante de un incremento en la probabilidad de conjugación de las transconjugadas, lo que hace que aumente el número de transconjugadas y a su vez de donantes, ya que las transconjugadas se convierten en donantes pasados cinco tiempos de generación. Este proceso causará un aumento en los datos de ambos tipos de bacterias, lo que hará que este último gráfico se aproxime con mayor precisión a los datos del IBBTEC.

Esta versión es más consecuente con lo que ocurre biológicamente, dado que las bacterias solo intentan conjugarse una sola vez debido al esfuerzo que conlleva este proceso para ellas, pero dado que aún queda por implementar el proceso *overshooting*, se consiguen datos más aproximados con la versión simplificada por las razones explicadas en el anterior apartado.



### 6.3 Conclusiones

En este proyecto se ha añadido la funcionalidad de comunicación intercelular mediante conjugación al simulador GRO, además de validar experimentalmente su funcionamiento con datos experimentales del IBBTEC [13].

GRO es un simulador multicelular basado en individuos que interaccionan entre sí. Actualmente, además de todas las funcionalidades que ya poseía, tiene la capacidad de representar el proceso de comunicación intercelular mediante conjugación. Se ha demostrado que ha sido posible aproximar la dinámica de este proceso a ciertos datos experimentales, quedando a la espera de la implementación del proceso *overshooting*. Este proceso hará que los datos sean más aproximados debido a que las bacterias transconjugadas sufren un incremento en su probabilidad de conjugación durante cinco tiempos de generación.

Sin embargo, se han integrado en GRO las tasas de crecimiento de cada tipo de bacteria, así como la ralentización que sufren las bacterias transconjugadas durante 3 tiempos de generación, lo que hace que en el tiempo en que las bacterias donantes y las receptoras se han dividido 3 veces aproximadamente, las transconjugadas se habrán dividido una sola vez. Pero, en el momento que se dividen por primera vez, recobran las tasas normales de crecimiento.

El proceso de conjugación se ha modelado eligiendo qué bacteria va a conjuguar según una probabilidad de conjugación, la cual depende de la tasa de conjugación  $\Gamma_0$ , que se da como parámetro al programa, y el número de iteraciones computacionales que hará antes de dividirse. En el momento en que una célula se ha elegido para conjuguar, se halla su lista de vecinas creando un aura de la bacteria (duplicando su radio) y eligiendo una bacteria al azar entre las bacterias con las que colisiona.

Se han creado dos versiones dependiendo de si la bacteria elegida al azar es Donante, Transconjugada o Receptora. La versión simplificada solo cuenta como conjugación efectiva aquella que se ha realizado a una Receptora, por lo tanto, si elige una bacteria que no es Receptora volverá a intentarlo con otra en la siguiente iteración. Por otro lado, en la versión completa, si la bacteria elegida no es Receptora cuenta como conjugación efectiva y la bacteria que va a conjuguar pierde su oportunidad de transmitir un plásmido a una Receptora.

Comparando ambas versiones, en la versión completa se consigue una visión mas precisa de lo que ocurre biológicamente, aunque en este momento, avanzada una simulación, pueda ser más precisa la versión simplificada por el hecho de que ocurren más conjugaciones efectivas (Donante-Receptora) y se aproxima a lo que ocurriría con el proceso de *overshooting*.

En conclusión, se ha conseguido modelar y parametrizar el proceso de conjugación y añadirlo al simulador GRO, además de ajustar las tasas en cuanto al crecimiento de los tres distintos tipos de bacterias. Ambos puntos han sido verificados con datos experimentales y se ha concluido que, a falta de mejoras tales como el proceso *overshooting*, se consigue representar este proceso de forma satisfactoria. Esta nueva funcionalidad añadida a GRO permitirá simular circuitos genéticos sintéticos multicelulares como los desarrollados en el proyecto europeo PLASWIRES.

### 6.4 Trabajos Futuros

A continuación, se mencionan algunos puntos sobre las posibles mejoras sobre este modelo:

- Añadir el proceso *overshooting*.
- Definir el método de conjugación como función externa para poder utilizarla como parte del lenguaje *gro*.
- Definir un mecanismo de ajuste de datos semi-automático, en el se indicaría la evolución de los Tiempos de Generación de cada tipo de bacteria y GRO se ajustaría a ellos.
- Estudio de distintas distribuciones de la probabilidad de conjugación
- Estudio de cómo afecta en la simulación de conjugación que la bacteria sea un poco mas grande.

## REFERENCIAS

- [1] Goñi-Moreno A, Amos M, de la Cruz F (2013) *Multicellular Computing Using Conjugation for Wiring*. PLoS ONE 8(6): e65986. doi:10.1371/journal.pone.0065986
- [2] Timothy J. Rudge, Paul J. Steiner, Andrew Phillips, and Jim Haseloff. (2012). *Computational Modeling of Synthetic Microbial Biofilms*. dx.doi.org/10.1021/sb300031n | ACS Synth. Biol. 2012, 1, 345–352
- [3] Gorochoowski TE, Matyjaszkiewicz A, Todd T, Oak N, Kowalska K, et al. (2012) *BSim: An Agent-Based Tool for Modeling Bacterial Populations in Systems and Synthetic Biology*. PLoS ONE 7(8): e42790. doi:10.1371/journal.pone.0042790
- [4] Jan-Ulrich Kreft, Ginger Booth and Julian W. T. Wimpenny. (1998). *BacSim, a simulator for individual-based modelling of bacterial colony growth*. doi:10.1099/00221287-144-12-3275 Microbiology December 1998 vol. 144 no. 123275-3287
- [5] Angel Goñi-Moreno and Martyn Amos. (2009). *DiSCUS: A Simulation Platform for Conjugation Computing*.
- [6] Antonio P. García. Tesis del máster en Inteligencia Artificial. (2011). *A first approach to individual-based modeling of the bacterial conjugation dynamics*.
- [7] Laurent A. Lardon, Brian V. Merkey, Sónia Martins, Andreas Dötsch, Cristian Picioreanu, Jan-Ulrich Kreft and Barth F. Smets. *iDynoMiCS: next-generation individual-based modelling of biofilms*. doi:10.1111/j.1462-2920.2011.02414.x
- [8] Arteaga, J.P. Tesis del máster en Inteligencia Artificial. (2012). *Enhancing iDynoMiCS framework to simulate Rod-shape bacterial colonies growth*.
- [9] Aurora S. Arroyo García. Tesis del máster en Inteligencia Artificial. (2013). *Enhancing iDynoMiCS framework with a plasmid conjugation module*.
- [10] Seunghye S. Jang, Kevin T. Oishi, Robert G. Egbert, and Eric Klavins. (2012). *Specification and Simulation of Synthetic Multicelled Behaviors*. doi:10.1021/sb300034m | ACS Synth.
- [11] Diego Pardilla Mata. Tesis del máster en Inteligencia Artificial. (2011). *Estudio de protocolos de comunicación bacterianos: conjugación y quorum sensing*.
- [12] Juan Alvarez Bravo. (1991). *Mapa Bacteriológico de Bacilos Gram Negativos*.
- [13] Fernandez-Lopez R, del Campo I, Revilla C, Cuevas A, de la Cruz F (2014) *Negative Feedback and Transcriptional Overshooting in a Regulatory Network for Horizontal Gene Transfer*. PLoS Genet 10(2): e1004171. doi:10.1371/journal.pgen.1004171
- [14] Irene del Campo, Raúl Ruiz, Ana Cuevas, Carlos Revilla, Luis Vielva, Fernando de la Cruz. *Determination of conjugation rates on solid surfaces*. doi:10.1016/j.plasmid.2012.01.008
- [15] Angel Goñi-Moreno and Martyn Amos. (2012). *Discrete modelling of bacterial conjugation dynamics*. arXiv:1211.1146v1 [cs.MA]

Este documento esta firmado por



<b>Firmante</b>	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
<b>Fecha/Hora</b>	Fri Jun 06 21:44:36 CEST 2014
<b>Emisor del Certificado</b>	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
<b>Numero de Serie</b>	630
<b>Metodo</b>	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)